
Identifiable Feature Learning for Spatial Data with Nonlinear ICA

Hermann Hälvä¹

Jonathan So²

Richard E. Turner²

Aapo Hyvärinen¹

¹Department of Computer Science, University of Helsinki, Finland,

²Department of Engineering, University of Cambridge, UK

Abstract

Recently, nonlinear ICA has surfaced as a popular alternative to the many heuristic models used in deep representation learning and disentanglement. An advantage of nonlinear ICA is that a sophisticated identifiability theory has been developed; in particular, it has been proven that the original components can be recovered under sufficiently strong latent dependencies. Despite this general theory, practical nonlinear ICA algorithms have so far been mainly limited to data with one-dimensional latent dependencies, especially time-series data. In this paper, we introduce a new nonlinear ICA framework that employs t -process (TP) latent components which apply naturally to data with higher-dimensional dependency structures, such as spatial and spatio-temporal data. In particular, we develop a new learning and inference algorithm that extends variational inference methods to handle the combination of a deep neural network mixing function with the TP prior, and employs the method of inducing points for computational efficacy. On the theoretical side, we show that such TP independent components are identifiable under very general conditions. Further, Gaussian Process (GP) nonlinear ICA is established as a limit of the TP Nonlinear ICA model, and we prove that the identifiability of the latent components at this GP limit is more restricted. Namely, those components are identifiable if and only if they have distinctly different covariance kernels. Our algorithm and identifiability theorems are explored on simulated spatial data and real world spatio-temporal data.

1 INTRODUCTION

Inferring semantically useful low-dimensional latent features from high dimensional real world data is a central goal in unsupervised feature learning. Many well-established approaches such as independent component analysis (ICA) and dictionary learning employ linear transformations, whereas recent approaches, such as nonlinear versions of ICA (Hyvärinen et al., 2023), and variational autoencoders (Kingma and Welling, 2014), typically assume that the transformation from the latent to the observed space is highly nonlinear, often modeled as a deep neural network.

Moving from linear to nonlinear generative models is likely to provide superior performance in many tasks, but also comes with challenges in interpretability and estimation. On the interpretability side, the problem is that many of these models are not *identifiable* – that is, they do not discover the ground-truth latent features, even in the limit of infinite data (Hyvärinen and Pajunen, 1999; Locatello et al., 2019). Without access to the ground-truth latent features, we can expect worse performance and difficulties in interpreting results in several important tasks such as classification (Klindt et al., 2021; Banville et al., 2021), transfer learning (Khemakhem et al., 2020b), and causal inference (Monti et al., 2019; Wu and Fukumizu, 2020). In order to overcome these challenges, there has been an increased interest in developing identifiable generative models and the related theory. Much of this work has been in the context of nonlinear ICA as it would provide a way to perform *principled* disentanglement learning, that is, without their usual problem of unidentifiability (Locatello et al., 2019; Hyvärinen et al., 2023).

The central approach taken in such works has been to provide inductive biases that allow the models to be identified. The type of inductive biases used can be roughly divided into two groups. The first group assumes additional observed ‘auxiliary’ data (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2019; Khemakhem et al., 2020a), such as audio that goes with the video data, or explicit knowledge about the time points at

which non-stationary process switches distributions. Such auxiliary data is not always available, however. To cope with this, Hyvärinen and Morioka (2017); Hälvä and Hyvärinen (2020); Hälvä et al. (2021) instead assume *purely latent* structures that allow identifiability. These latent structures have been primarily different types of temporal dependencies, with applications on time-series data such as brain imaging. The only work that goes beyond temporal dependencies seems to be Hälvä et al. (2021) who provide general theorems with *sufficient* conditions that allow model identifiability with arbitrarily high dimensional latent dependencies, such as spatial data. Nevertheless, their work is mainly theoretical as the algorithm in Hälvä et al. (2021) only applies to time-series data. In fact, no unsupervised learning method exists for performing non-linear ICA on data with more complex latent dependencies. This is a severe limitation as it hinders the applications of nonlinear ICA on many domains such as spatial data, where one would expect 2D dependencies, or spatio-temporal with 3D latent structures. These types of dependencies are ubiquitous, for example, in remote sensing and data used in meteorological, earth and other environmental sciences.

A well-known approach to modelling spatial dependencies would be to use Gaussian Processes (GP). However, the importance of non-Gaussianity is well-known in the linear ICA literature; whether this is the case in nonlinear ICA has not been thoroughly studied until now however. In this paper, we consider this via Student’s t-Processes (TP) (Yu et al., 2007; Shah et al., 2014) of which GPs are a special case. In particular, we introduce Student’s t-process nonlinear ICA (tp-NICA) which resolves the aforementioned limitations of temporal models by assuming TP latent components. Most importantly, unlike previous works, our model and the accompanying algorithm allow arbitrarily high dimensional latent dependencies. This makes the model particularly attractive for disentangling identifiable features from spatial data such as images or geographically structured data, as we will demonstrate in our experiments.

Thus, the first major contribution of our paper is extending nonlinear ICA algorithms beyond time-series to spatial and other data with high-dimensional dependencies, without necessitating any additional auxiliary data. We also show the identifiability of the model by adapting the theory of Hälvä et al. (2021). Further, we establish GP nonlinear ICA (gp-NICA) as the limit of the tp-NICA model, but show that its identifiability is more constrained: the gp-NICA model is identifiable if and only if all the different processes have unique covariance kernels.

Our second major contribution is a new learning and in-

ference algorithm for our tp-NICA model. In particular, our algorithm shows how to overcome two fundamental challenges: i) intractability that results from a non-conjugate non-exponential family latent prior combined with a nonlinear, neural network, observation likelihood, and ii) the expensive, cubic, computational cost typically associated with GPs and TPs. Our algorithm resolves these challenges by constructing a variational lower bound that utilizes the infinite GP-mixture representation of TPs, and by employing pseudo-points from sparse GP literature for computational efficacy, as well as other advances in variational inference.

2 BACKGROUND

In this section we provide a brief technical background of some of the key concepts that we employ in our tp-NICA model and our learning and inference algorithm.

2.1 Identifiability and Nonlinear ICA

A probabilistic model is considered identifiable if it is possible, at least in theory, to learn the model’s ground-truth parameters. An unidentifiable model fails to satisfy this condition, and thus, any parameter estimates for such a model may not correspond to the ground-truth parameters. Formally, a model is identifiable up to some equivalence relation \sim if $p_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\theta}) = p_{\mathbf{x}}(\mathbf{x}; \tilde{\boldsymbol{\theta}}) \implies \boldsymbol{\theta} \sim \tilde{\boldsymbol{\theta}}$, where $p_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\theta})$ denotes the probability distribution of a random variable \mathbf{x} parameterized by $\boldsymbol{\theta}$. Importantly for the present work, the most basic formulation of nonlinear ICA, namely a factorial prior on some latent components $p(\mathbf{s}) = \prod_{i=1}^N p(s^{(i)})$ along with a nonlinear mixing function $\mathbf{x} = \mathbf{f}(\mathbf{s})$, has been shown to be catastrophically unidentifiable (Hyvärinen and Pajunen, 1999) – an infinite number of possible solutions exist. Several successful advances have been made over the recent years in developing alternative, identifiable, nonlinear ICA models. Of these, the most relevant to ours is Hälvä et al. (2021), whose Structured Nonlinear ICA (SNICA) framework defines a general set of sufficient identifiability conditions. These also apply to our model, and as such we will discuss them here in more detail. The SNICA framework utilizes a noisy version of the classic nonlinear ICA model

$$\mathbf{x}_l = \mathbf{f}(\mathbf{s}_l) + \boldsymbol{\varepsilon}_l \quad \forall l \in \mathbb{L}, \quad (1)$$

where \mathbb{L} is an indexing set that could be arbitrarily high-dimensional (e.g. 2 dimensional for spatial data). Hälvä et al. (2021) provide two main identifiability results. First, the noise-free mixture $\mathbf{f}(\mathbf{s}_l)$ can be identified from the noisy model, for arbitrary noise in (1), under

mild conditions that relate to the behaviour and non-Gaussianity of the noise-free mixture. Second, the ground-truth de-mixing function \mathbf{f}^{-1} can be obtained from the distribution of $\mathbf{f}(\mathbf{s}_i)$ up to permutation and coordinate-wise transformations as long as conditions that govern sufficiently strong statistical dependency between index locations, and the condition of non quasi-Gaussianity, are satisfied. Hälvä et al. (2021) also proposed a specific practical algorithm based upon their framework where the latent components follow autoregressive and hidden Markov models. While it was shown to be successful for temporal data, it cannot be applied on spatial data in any reasonable way, which is our main concern in this paper.

2.2 Gaussian Processes and t -Processes

Gaussian process (GP) is a stochastic processes that specifies the joint distribution of infinitely many random variables such that any finite sample of those variables obeys a multivariate Gaussian distribution. Since GPs can represent the joint-distribution of infinitely many random variables, they are used as priors for continuous functions as well as latent processes. The dependencies between any variables in a GP are captured by a kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$, where \mathbf{x}_i represents some input variable, such as location in space. For temporal data these could be one dimensional indices, while for spatial data they could, for example, be locations on a lattice. The crucial aspect, for this paper, is that the input variables can have an arbitrarily high dimension. GPs are also a limiting case of Student's- t Processes (TP), as we show next.

If $\mathbf{x} \in \mathbb{R}^d$ follows a multivariate- t distribution with $\nu > 0$ degrees of freedom, we write $\mathbf{x} \sim t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with its p.d.f given by

$$p(\mathbf{x}) = \pi^{-\frac{d}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \nu^{\frac{\nu}{2}} \frac{\Gamma(\frac{\nu+d}{2})}{\Gamma(\frac{\nu}{2})} \times (\nu + (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))^{-\frac{\nu+d}{2}}, \quad (2)$$

where $\Gamma(\cdot)$ is the Gamma function. When $\nu \rightarrow \infty$ this becomes a Gaussian p.d.f. Following Yu et al. (2007), we can define t -Processes as follows:

Definition 1 (*t -Process*). A random function $y : \mathbb{R}^d \rightarrow \mathbb{R}$ follows a t -Process ($\mathcal{TP}_\nu(h, \kappa)$) with $\nu > 0$ degrees of freedom, mean function $h(\cdot)$ and covariance kernel $\kappa(\cdot, \cdot)$, if for any $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ with a finite $n > 0$:

$$\mathbf{y} = (y(\mathbf{x}_1), \dots, y(\mathbf{x}_n))^T \sim t_\nu(\mathbf{h}, \mathbf{K}),$$

where $\mathbf{h} = (h(\mathbf{x}_1), \dots, h(\mathbf{x}_n))^T$ and $\mathbf{K}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$.

It thus follows that a GP is the limit of a TP as the degrees of freedom parameter tends to infinity.

It is also well-known that a t -distribution can be re-framed as infinite mixture of Gaussian distributions with each Gaussian's covariance scaled by a Gamma distributed random variable. This extends also to \mathcal{TP} s as per the following proposition (Yu et al., 2007):

Proposition 1 (*\mathcal{TP} as infinite mixture*). $\mathcal{TP}_\nu(h, \kappa)$ can be sampled by repeatedly sampling a Gamma distributed random variable $\tau \sim \text{Gamma}(\frac{\nu}{2}, \frac{\nu}{2})$ followed by $\varphi \sim \mathcal{GP}(h, \frac{1}{\tau}\kappa)$.

2.3 Variational Inference

Our inference and learning algorithm combines several advances in variational inference and sparse GP literature and thus we provide a short background here. In the factored variational approach the posterior over a latent \mathbf{y} is assumed to factorize over some partition $\mathcal{Z}_{1:M}$ such that $q(\mathbf{y}) := \prod_{j \in \mathcal{Z}_m} q_j(\mathbf{y}_j)$ is tractable, which allows a variational lower bound to be computed and optimized. Such approximations do not typically work for models with complex observation likelihoods such as neural networks. In black-box variational inference (BBVI) (Ranganath et al., 2014) an approximate posterior distribution $q(\mathbf{y}_i | \boldsymbol{\lambda}_i)$ with free-form variational parameters $(\boldsymbol{\lambda})_{i=1, \dots, m}$ is optimized. Each observation has its own free-form variational parameters, thus allowing for inference in complex distributions. The high variance of the gradients in BBVI is problematic however; Rezende et al. (2014); Kingma and Welling (2014) instead introduced the reparametrization trick which provides lower-variance gradients with respect to the variational parameters $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Further, to deal with the intractable posterior $q(\mathbf{y})$, VAEs (Kingma and Welling, 2014) approximate it with a diagonal Gaussian: $\mathbf{y}_i | \mathbf{x}_i \sim \mathcal{N}(\mathbf{m}(\mathbf{x}_i), \boldsymbol{\sigma}(\mathbf{x}_i)^2 \mathbf{I})$ where $\mathbf{m}(\cdot), \boldsymbol{\sigma}(\cdot)$ are the outputs of an encoder neural network. Since VAEs assume factorial posteriors, they are not suited for models with structured latent dependencies. Johnson et al. (2016) resolve this by assuming an approximate posterior in the form of $q(\mathbf{y}) \propto p(\mathbf{y}) l_\lambda(\mathbf{x} | \mathbf{y})$, where $l_\lambda(\mathbf{x} | \mathbf{y})$ is an approximate likelihood term chosen to be conjugate to the desired structured exponential family prior (such as a HMM). The approximate likelihood is again formed by an encoder neural network. This approach has inspired deep generative models with GP priors (see Casale et al. (2018); Pearce (2020); Ashman et al. (2020); Jazbec et al. (2021)). For instance, Ashman et al. (2020) assume an approximate posterior:

$$q(\mathbf{y}^{(i)}) \propto p(\mathbf{y}^{(i)}) \prod_{t=1}^m \psi(\mathbf{x}_t^{(i)}; \mathbf{y}^{(i)}, \mathbb{L}) \quad (3)$$

with $q(\mathbf{y}) = \prod_i q(\mathbf{y}^{(i)})$, where $\mathbf{y}^{(i)} \sim \mathcal{GP}^{(i)}$, $(\mathbf{x}_m^{(i)})_{\forall m}$ are observations at m locations of the indexing set

\mathbb{L} , where the approximate likelihood $\psi(\cdot)$ takes Gaussian form. Naive implementations of GPs suffer from cubic computational cost. Sparse GPs address this with the so called pseudo-points (Snelson and Ghahramani, 2005; Quiñero-Candela and Rasmussen, 2005) which essentially summarize the GP at a much smaller number of pseudo-locations. These locations can be learned in a variational framework (Titsias, 2009; Hensman et al., 2013, 2015). In practice many of the GP-VAE methods discussed above utilize these pseudo-points. For instance, (3) in Ashman et al. (2020) become $q(y^{(i)}) \propto p(y^{(i)}) \prod_{t=1}^L \psi(\mathbf{x}_t^{(i)}; \mathbf{u}, \mathbf{Z}, \mathbb{L})$ where \mathbf{u} are pseudo-observations at pseudo-locations \mathbf{Z} and $L \ll m$.

3 MODEL DEFINITION

To define our proposed model, t -Process nonlinear ICA (tp-NICA), we start by assuming there exist N statistically independent latent features $\mathbf{s}^{(i)} = (s_l^{(i)})_{l \in \mathbb{L}}$ for $i \in \{1, \dots, N\}$. Here \mathbb{L} is an indexing set of arbitrary dimension. While previous theoretical works have sometimes allowed \mathbb{L} to also have any dimension, practical NICA algorithms have only allowed for one-dimensional indices, temporal index in particular. Here, our main application areas are spatial and spatio-temporal data and hence \mathbb{L} will usually be two- or three-dimensional, in other words, it's a subset of \mathbb{N}^2 or \mathbb{N}^3 , so that we could equivalently write $\mathbf{s}^{(i)} = (s_l^{(i)})_{l \in \mathbb{N}^2} = (s_{k,l}^{(i)})_{k,l \in \mathbb{N}}$. Our algorithm will also be able to handle any other higher dimensional \mathbb{L} and thus is not limited to spatial or spatio-temporal data.

To allow for high-dimensional \mathbb{L} , we assume that each of the independent components follows a t -Process (TP) over the indexing set \mathbb{L} . TPs are a natural prior for independent components due to their non-Gaussianity and ability to model complex dependencies, but they have not previously been considered for NICA. Compared to GPs, they are also more flexible as the degree of freedom parameter controls the heaviness of the distribution's tails; allowing for heavy tails is useful as it is a characteristic that has been empirically observed in independent components (Hyvärinen et al., 2009). Thus, we define

$$\mathbf{s}^{(i)} \sim \mathcal{TP}_{\nu^{(i)}}(h^{(i)}, \kappa^{(i)}), \quad (4)$$

with $h : \mathbb{L} \rightarrow \mathbb{R}$ and $\kappa : \mathbb{L}^2 \rightarrow \mathbb{R}$. The joint distribution of the independent components is defined as $p(\mathbf{s}_{l_1}, \dots, \mathbf{s}_{l_m}) = \prod_{i=1}^N p(s_{l_1}^{(i)}, \dots, s_{l_m}^{(i)})$ for $m \in \mathbb{N}^*$ and $(l_1, \dots, l_m) \in \mathbb{L}^m$. Additionally, we assume that the observed data is created from the latent components by a nonlinear mixing function that operates at each index of \mathbb{L} so that $\mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ with $M \geq N$ is injective. We also assume observation noise denoted

by $\boldsymbol{\varepsilon}_l \in \mathbb{R}^M$ that is i.i.d across the M dimensions as well as with respect to the independent components. The noise can take any arbitrary distribution in theory but in practice we will usually assume a Gaussian distribution. As such, the observed data is generated as per Equation (1) at each index $l \in \mathbb{L}$. Note that this implies $p(\mathbf{x}_l | \mathbf{s}_l) = p_{\boldsymbol{\varepsilon}}(\mathbf{x}_l - \mathbf{f}(\mathbf{s}_l))$. Finally, the full joint distribution factorizes as

$$p(\mathbf{x}_{1:m}, \mathbf{s}_{1:m}) = \prod_{j=1}^m p(\mathbf{x}_{l_j} | \mathbf{s}_{l_j}) \prod_{i=1}^N p(\mathbf{s}^{(i)}) \quad (5)$$

where we denote $\mathbf{s}_{1:m} := s_{l_1}, \dots, s_{l_m}$.

Finally, if we take the degrees of freedom parameter ν in (4) to infinity, but otherwise keep the model as defined above, we establish the Gaussian Process NICA (gp-NICA) model in the limit and as a special case of the tp-NICA model

4 IDENTIFIABILITY

The tp-NICA model introduced above falls within the identifiability framework of Hälvä et al. (2021), thus leading to the following identifiability result:

Theorem 1 (*Identifiability of tp-NICA*) *Assuming that the assumptions (A1), (A2) and (A3) of Theorem 1 in Hälvä et al. (2021) apply¹, then the tp-NICA model is identifiable such that $p(\mathbf{x}; \mathbf{f}) = p(\mathbf{x}; \tilde{\mathbf{f}}) \implies \mathbf{f}^{-1} \sim \tilde{\mathbf{f}}^{-1}$, where \sim denotes equivalence up to permutation and coordinate-wise bijective transformation of the elements of the de-mixing function \mathbf{f}^{-1} .*

Our proof of this theorem (in Appendix B) relies on showing that t -processes are not (locally) quasi-Gaussian, which is a sufficient condition for identifiability (Hälvä et al., 2021). This however does not preclude the possibility that GP components could lead to identifiable models in some instances. Indeed, Hälvä et al. (2021) show that a noise-free variant of their model with GP independent components can be identified under the sufficient condition that the different GPs have distinct kernel functions. This raises the question whether non-Gaussianity is *necessary* for identifiability. Our Theorem 2 below shows that this is indeed the case. Subsequently, the gp-NICA model is identifiable if and *only if* each component has a distinct covariance kernel, which is in contrast to tp-NICA where such necessary conditions do not apply.

Theorem 2 (*Necessity of distinct covariance kernels with GP independent components*). *Assume that we have a model otherwise defined as in Section*

¹These assumptions are explained in Appendix B

3 except that in place of the \mathcal{TP} distributed components, the model has \mathcal{GP} distributed components: $\mathbf{s}^{(i)} \sim \mathcal{GP}(h^{(i)}, \kappa^{(i)}); i \in \{1, \dots, N\}$. Then $p(\mathbf{x}; \mathbf{f}) = p(\mathbf{x}; \tilde{\mathbf{f}}) \implies \mathbf{f}^{-1} \sim \tilde{\mathbf{f}}^{-1}$ if and only if the covariance kernels of the different components are unique so that $\kappa^{(i)}(s_l^{(i)}, s_m^{(i)}) \neq \kappa^{(j)}(s_l^{(j)}, s_m^{(j)}) \forall i \neq j$.

Note that Theorem 2 considers the noise-free case because it is more general. To see this, notice that any necessary conditions for the noise-free case must be satisfied also by the noisy case. Our proof (in Appendix B) follows directly by extending a related theory for linear ICA by Belouchrani et al. (1997).

5 LEARNING AND INFERENCE

The marginal log-likelihood of the tp-NICA model can be written as:

$$\mathcal{L} := \log p(\mathbf{x}) = \log \int_{\mathbf{s}} \prod_{j=1}^m p(\mathbf{x}_{l_j} | \mathbf{s}_{l_j}) \prod_{i=1}^N p(\mathbf{s}^{(i)}) d\mathbf{s}, \quad (6)$$

Maximization of this marginal log-likelihood is clearly intractable: $p(\mathbf{x}_l | \mathbf{s}_l)$ is a nonlinear observation likelihood parameterized by a deep neural network and Gaussian output noise, whilst $p(\mathbf{s}^{(i)})$ is a TP which is non-exponential family and thus non-conjugate, making it difficult to approximate $p(\mathbf{x}_l | \mathbf{s}_l)$ with a conjugate-likelihood (recall Section 2.3). Another challenge, as with GPs, is the apparent cubic computational cost.

Therefore, we make two alterations to (6). First, as a step towards tractability, recall Proposition 1. This allows us to write the TP priors as infinite mixtures of GPs scaled by gamma random variable τ . Second, we introduce additional pseudo-variables, \mathbf{u} , to control the computation complexity of our algorithm. These pseudo-variables can be considered as additional latent variables at some extra input locations (Section 2.3 and Appendix A.3). Importantly, note that the \mathbf{u} are *not* observed and thus no additional data is needed; they are merely mathematically convenient additional latent variables that we integrate out. With these additions we can compute a tractable lower bound to our new marginal likelihood (see Appendix A for all derivations):

$$\begin{aligned} \mathcal{L} &:= \log \int_{\tau} \int_{\mathbf{s}} \int_{\mathbf{u}} p(\mathbf{x}, \mathbf{s}, \mathbf{u}, \tau) d\mathbf{u} d\mathbf{s} d\tau \\ &\geq \mathbb{E}_{q(\tau)} \left[\mathbb{E}_{q(\mathbf{s}, \mathbf{u} | \tau)} [\log p(\mathbf{x} | \mathbf{s}, \mathbf{u})] \right. \\ &\quad \left. - \text{KL}[q(\mathbf{s}, \mathbf{u} | \tau) \| p(\mathbf{s}, \mathbf{u} | \tau)] - \text{KL}[q(\tau) \| p(\tau)] \right] \\ &\geq \mathbb{E}_{q(\tau)} \underbrace{\left[\mathbb{E}_{\tilde{q}(\mathbf{s} | \tau)} [\log p(\mathbf{x} | \mathbf{s})] \right]}_{(*)} \\ &\quad - \underbrace{\text{KL}[q(\mathbf{u} | \tau) \| p(\mathbf{u} | \tau)]}_{(**)} - \underbrace{\text{KL}[q(\tau) \| p(\tau)]}_{(***)}, \quad (7) \end{aligned}$$

where we denote $\tilde{q}(\mathbf{s} | \tau) := \int p(\mathbf{s} | \mathbf{u}, \tau) q(\mathbf{u} | \tau) d\mathbf{u}$. The second bound is due to our choosing to approximate $q(\mathbf{s} | \mathbf{u}, \tau) := p(\mathbf{s} | \mathbf{u}, \tau)$, which allows for the computational benefits of using the pseudo-points. Further, we choose the following non-factorial posterior approximation for the pseudo-latents:

$$\begin{aligned} q(\mathbf{u}, \tau) &= q(\mathbf{u} | \tau; \phi, \theta) q(\tau; \phi) \\ &:= \frac{\prod_{j=1}^J \psi(\mathbf{u}_j; \phi_j) \prod_{i=1}^N p(\mathbf{u}^{(i)} | \tau^{(i)}; \theta^{(i)})}{Z(\phi, \theta, \tau)} \quad (8) \\ &\quad \times \prod_{i=1}^N q(\tau^{(i)}; \phi^{(i)}), \end{aligned}$$

where J is the number of pseudo-locations, $\mathbf{u}_j = (u_{l_j}^{(i)})_{i=1:N}$, and $\mathbf{u}^{(i)} = (u_{l_j}^{(i)})_{j=1:J}$. Crucially, separating the TP prior into Gamma and \mathcal{GP} distributions has now allowed us to approximate the previously intractable pseudo-likelihood term with a conjugate Gaussian factor $\psi(\mathbf{u}_j; \phi_j) = \exp \left\{ -\frac{1}{2} (\mathbf{u}_j^T \tilde{\mathbf{W}}_j^T \tilde{\mathbf{W}}_j \mathbf{u}_j - 2\mathbf{u}_j^T \tilde{\mathbf{m}}_j) \right\}$, where $(\tilde{\mathbf{m}}_j, \tilde{\mathbf{W}}_j) \in \phi_j$ are free-form variational parameters akin to Ranganath et al. (2014), in particular $\tilde{\mathbf{W}}$ are parameterized as Cholesky factors. Notice that in (8) the approximate Gaussian factor and $p(\mathbf{u}^{(i)} | \tau^{(i)}; \theta^{(i)})$ factorize over different indices, and consequently the resulting Gaussian $q(\mathbf{u} | \tau; \phi, \theta)$ has a sparse, but non-factorized, form, which is in contrast to previous works that assume various limiting factorizations of the posterior.

The final lower bound in (7) allows sufficient tractability to estimate our model parameters and to perform inference. Terms (**), (***) can be computed in closed-form as they are just KL divergence between \mathcal{GP} and Gamma distributions, respectively. The data-likelihood denoted (*) is parameterized by a decoder network and thus remains intractable so that we take the VAE approach of reparametrized gradients after sampling from $\tilde{q}(\mathbf{s} | \tau)$ which is a Gaussian. Finally, we also take reparametrized gradients of the outer expectation over the Gamma distributed $q(\tau^{(i)}; \phi^{(i)})_{\forall i}$. In practice, we use autograd to take gradients through all of (7) and then optimize all the model parameters θ and the free-form variational parameters (ϕ, \mathbf{Z}) with stochastic gradient descent, where \mathbf{Z} are locations of the pseudo-latents. A step-by-step description is detailed in Algorithm 1. For gp-NICA the Gamma expectation and KL-divergence terms in equation (7) disappear, but otherwise the same algorithm can be utilized.

6 EXPERIMENTS

In this section we explore the performance of the tp-NICA and gp-NICA models on simulated spatial as well as real world data. Our code will be openly available

Algorithm 1 Estimate tp-NICA lower bound and take gradients

Require: model parameters $\theta := (\theta^{(i)})_{i=1,\dots,N}$, variational parameters $\phi := ((\phi^{(i)})_{i=1,\dots,N}, \mathbf{Z})$, data \mathbf{x} , number of posterior τ samples N_t , number of posterior \mathbf{s} samples N_s

function ELBOGRADIENTS($\theta, \phi, \mathbf{x}, N_t, N_s$)

$\{\hat{\tau}_t\}_{t=1:N_t} \overset{i.i.d.}{\sim} q(\tau; \phi)$ $\triangleright N_t$ Samples from approximate posterior

$\{\mathcal{L}_t\}_{t=1:N_t}, \{\hat{\mathbf{S}}_t\}_{t=1:N_t} \leftarrow \text{CONDITIONALINFERENCE}(\{\hat{\tau}_t\}_{t=1:N_t}, \theta, \phi, \mathbf{x}, N_s)$

$\mathcal{L} \leftarrow \frac{1}{N_t} \sum_{t=1}^{N_t} \mathcal{L}_t - \text{KL}[q(\tau; \phi) \parallel p(\tau, \gamma)]$ \triangleright Approximates (7)

$\nabla_{\theta} \mathcal{L}, \nabla_{\phi} \mathcal{L} \leftarrow \text{Autograd}(\mathcal{L}, \theta, \phi)$ \triangleright Compute gradients with re-parametrization trick

return $\mathcal{L}, \nabla_{\theta} \mathcal{L}, \nabla_{\phi} \mathcal{L}, \{\hat{\mathbf{S}}_t\}_{t=1:N_t}$

end function

function CONDITIONALINFERENCE($\hat{\tau}_t, \theta, \phi, \mathbf{x}, N_s$)

$\{\hat{\mathbf{s}}_s\}_{s=1:N_s} \overset{i.i.d.}{\sim} \tilde{q}(\mathbf{s} \mid \hat{\tau}_t; \theta, \phi)$ $\triangleright N_s$ samples as per Sec. A.5

$\mathcal{L}_t \leftarrow \frac{1}{N_s} \sum_{s=1}^{N_s} \log p(\mathbf{x} \mid \hat{\mathbf{s}}_s) - \text{KL}[q(\mathbf{u} \mid \hat{\tau}_t; \theta) \parallel p(\mathbf{u} \mid \hat{\tau}_t; \phi)]$ \triangleright KL as per Sec A.6

return $\mathcal{L}_t, \hat{\mathbf{S}}_t$ $\triangleright \hat{\mathbf{S}}_t := \{\hat{\mathbf{s}}_s\}_{s=1:N_s}$

end function

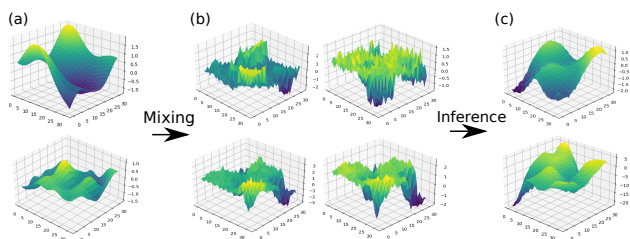


Figure 1: A simple illustration of (a) two TP independent components, (b) their four-dimensional mixture + observation noise (c) the inferred components at the end of tp-NICA training with $\text{MCC}=0.95$. Three mixing layers were used. Comparison of (a) and (c) illustrates well how the recovered components are identifiable up to component-wise bijective transformations.

at <https://github.com/cambridge-mlg/tp-nica>.

6.1 Experiments on Simulated Data

We first tested how well the algorithms are able to estimate the ground-truth latent components from simulated spatial data. Since the identifiability theorems hold in the limit of infinite data and universal consistent estimators, it is of interest to see how these theorems are obeyed in practice in scenarios where the data and the estimator methods stray away from such theoretical ideals. For the simulation, we sampled spatial 32×32 latent components on a regular lattice, using a squared-exponential kernel. Both tp-NICA and gp-NICA were estimated on their own data. We considered specifications with 6 and 12 latent components, which were mixed into 12 and 24 32×32 observations, respectively, using a mixing neural network. The depth

of the neural network was varied from one to four layers to simulate different levels of nonlinearity (one layer of mixing corresponds to linear ICA). Two data sets were generated, one where the squared exponential kernels’ parameters were engineered to be distinctly different for each component, and another where the kernels for all the components were forced to be the same. Figure 1 illustrates the type of data generated and displays an example of successful demixing by the tp-NICA model.

Identifiability was measured by the mean absolute correlation coefficient (MCC) between the ground truth and estimated independent components. Recall that our theorems show that the model parameters are identifiable up to permutation of the components and component-wise bijective transformation – both of these present problems to using the MCC as a performance metric. First, to deal with the arbitrariness in the components’ ordering, the ground-truth and the estimated components were matched to each other using a linear sum assignment algorithm (Crouse, 2016) prior to computing the MCC. Second, since each component is identifiable up to a bijective transformation, a Pearson correlation is not appropriate and we instead use Spearman rank correlation to compute the MCC.

None of the previous NICA algorithms are able to exploit spatial dependencies so there is no natural competitive benchmark. Nevertheless, we have included linear ICA and iVAE (Khemakhem et al., 2020a) as naive baselines. These both models treat the spatial locations as independent. Further, iVAE requires some observed auxiliary data, which here we have taken to be the locations of the latent variables (i.e. the same input as into the GP kernels).

Our results in Figure 2 largely correspond to theory: For data where kernels are forced to have distinctly

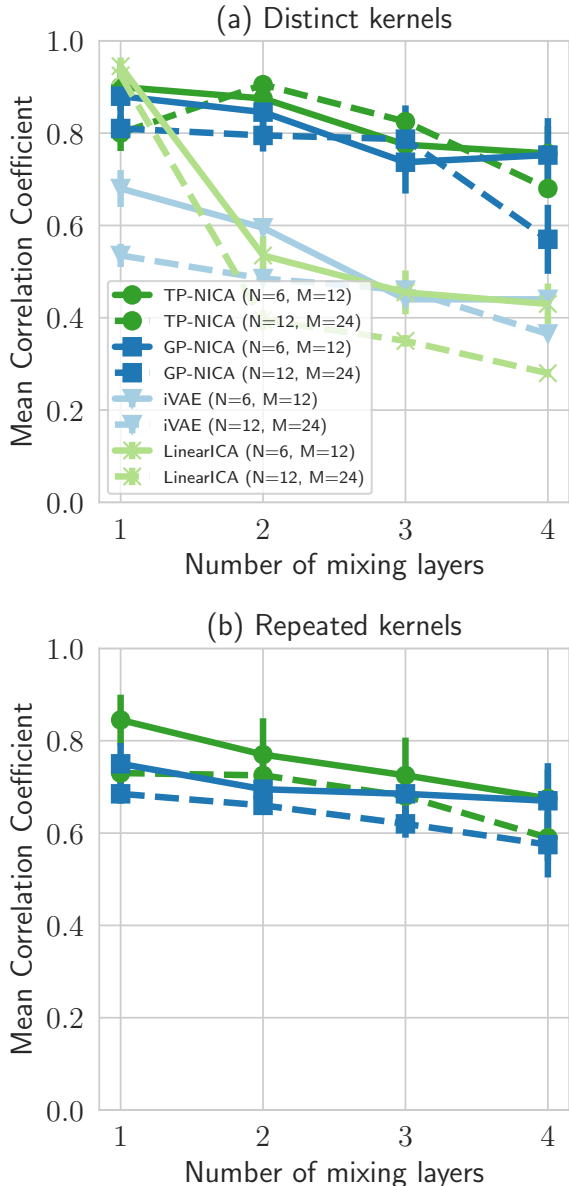


Figure 2: Mean absolute correlation coefficients between ground-truth independent components and their estimates for tp, gp-NICA, and iVAE and linear ICA baselines. gp-, tp-NICA are estimated on their own data, while the baseline models were estimated on data from both models with nearly identical results. Results shown for varying levels of non-linearity (number of layers) for (a) distinctly different covariance kernels between the components, and (b) components with equivalent kernels. The error bars show standard deviation from five different random seeds.

different parameters, both tp-NICA and gp-NICA perform well, and much better than the naive baselines. The performance deteriorates slowly for all models as nonlinearity increases, which is to be expected due to limited data, computation, and estimation methods. For the setting with equal kernels across components, tp-NICA has 0.05-0.15-points higher MCC, showing that it’s indeed more identifiable in this scenario as theory predicts. However, in contrast to theory, the gp-NICA model is not completely unidentifiable and as the number of mixing layers increases the two models performance converges. This is likely because in practice constraints such as the use of specific kernel and specific MLP mixing function imposes inductive biases which is in contrast to the theoretical assumptions of arbitrary mixing function and arbitrary covariance kernels. Also, in the most nonlinear setting, the difficulty of estimating the mixing likely dominates the results, rather than identifiability. Further implementation details and additional simulated experiments are found in the Appendix.

6.2 Experiments on Real Spatio-temporal Data

We also tested our models on real spatio-temporal data. In particular, we used data from the Computer Vision for Agriculture (CV4A) Kenya Crop Type dataset (RadiantEarthFoundation, 2020). This dataset was originally designed for an entirely different supervised classification task, but we adapted it into a suitable test environment for our model. Our modified dataset contains Sentinel-2 L2A satellite imagery of over 4000 fields of crop of 10m spatial resolution, processed to 32x32 images. The image data from the satellite is multi-spectral (16 dimensions) and for each field, covers the growing season from early June until early August (6 time-points for each field). This type of data can be extremely valuable in helping farmers increase productivity via scalable agricultural monitoring as well as helping to reach sustainability goals (RadiantEarthFoundation, 2020).

We first trained the tp-NICA model in a fully *unsupervised* fashion on this data. To exploit both spatial and temporal dependencies over the crop growth-cycle, each independent component was taken to be spatio-temporal: 6 time-points (avg. 10 days apart) over the 32x32 spatial grid. 100 inducing points and the squared-exponential kernel were used. The inferred independent components were evaluated on a supervised classification task of finding the correct temporal ordering of the satellite images; that is ~ 4000 satellite images from all of the 6 time-points were classified into those 6 time-points based only on the learned independent components using a Random Forest (RF) classifier. Figure

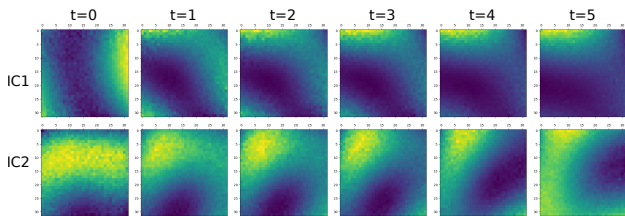


Figure 3: Examples of two nonlinear independent components inferred from the CV4A data. Horizontal axis shows how the components evolve over a field’s growth cycle.

3 depicts the type of spatio-temporal dependencies that the tp-NICA model learned. The results are displayed in Table 1, based on 10-fold cross-validation, to ensure generalization. The results show that tp-NICA and gp-NICA perform much better than the baseline RF classifier on the original image data. These results highlight the challenge of this task – a successful classification requires that the learned features contain salient temporal and spatial structure that is shared across majority of the images. We also found that tp-NICA performed consistently better than gp-NICA, likely due to its ability to model any fatter tails in the data; indeed, tp-NICA models with small d.o.f. parameters performed the best (see Appendix). These results are very promising as remote agricultural monitoring of crops’ growth cycles could have a massive benefit to farmers.

7 RELATED WORK

Unlike previous works, tp-NICA is built to exploit arbitrarily high-dimensional latent dependencies. It is, in particular, the first nonlinear ICA model that is built for analysing spatial and spatio-temporal data. Nevertheless, it shares several features with, and has been inspired by, several previous works.

The typical approach in existing nonlinear ICA works has been to exploit various inductive biases and latent dependencies in order to show model identifiability. One approach to this has come in the form of additional auxiliary information (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2019; Khemakhem et al., 2020a). It could be argued that the tp-NICA and gp-NICA models fall under this category as the covariance kernels take as input the spatial indices, which could thus be seen as auxiliary data. A critical difference is, however, that this auxiliary data is obvious and readily available.

Another approach in nonlinear ICA has been to identify models using their latent dependencies: Hälvä and

Hyvärinen (2020) used a HMM of the latent components, while Hyvärinen and Morioka (2017); Schell and Oberhauser (2023) exploited general temporal dependencies. A model using very general dependencies was proposed by Hälvä et al. (2021), whose SNICA identifiability framework also subsumes our model. Despite the generality of their identifiability theory, their practical SNICA algorithm only applies to time-series or other one-dimensional dependencies.

In terms of identifiability theory, we established the *necessary* condition that GP latent components are identifiable *only if* the covariance kernels of the components are distinct. This is a fundamental result as it was previously unclear whether the *sufficient* conditions in Hälvä et al. (2021) about the distinctness of the kernels were necessary or whether perhaps GPs were identifiable even more generally. This result follows rather directly from Belouchrani et al. (1997).

On the algorithmic side, our work is most similar to Ashman et al. (2020); Jazbec et al. (2021) who define GP latent components with a neural network likelihood function. Like those authors, we employed the ideas of Hensman et al. (2013, 2015) on inducing points for computational scalability. In fact, their models are almost equivalent to the gp-NICA benchmark we used in our experiments. The tp-NICA model differs from these works in that by assuming t-process latent components, we are able to perform nonlinear ICA with components that are more generally identifiable. Also, our use of the t-process leads to a very different inference algorithm as TPs are not in the exponential family, and thus non-conjugate, hence requiring a novel approach; we solved this by re-phrasing our objective in terms of infinite mixture of GPs scaled by a Gamma random variable. Furthermore, unlike those two papers, we used free-form variational parameters (Ranganath et al., 2014) rather than amortizing our inference; we utilize the reparametrization trick in optimizing these free-form parameters, thus combining the strengths of BBVI (Ranganath et al., 2014) and VAE (Kingma and Welling, 2014) approaches. Finally, unlike Ashman et al. (2020); Jazbec et al. (2021) and most other previous works in VAEs, we do not assume a factored posterior approximation, but rather assume that the approximate likelihood term only factors over time but not between components – this allows us to parameterize it in terms of block diagonal matrix of Cholesky factors, where blocks size equals the number of components. Finally, we mention that Student-t distribution was previously considered in linear ICA by Tipping and Lawrence (2005).

Limitations Because of the non-factorial posterior, the computational complexity of our full algorithm

Table 1: Average cross-entropies and accuracies from Random Forest (RF) classification of the images into temporal order based on the learned independent components for tp- and gp-NICA. Baseline of classifying the actual images without feature extraction (RF only) is given. l=2 and l=3 indicate the number of mixing layers assumed in the model. Standard deviations across the 10-fold cross-validation are displayed.

	Random baseline	RF only	gp-NICA (l=2) + RF	gp-NICA (l=3) + RF	tp-NICA (l=2, nu=4) + RF	tp-NICA (l=3, nu=4) + RF
Avg. cross-entropy	1.81 ± 0.01	1.33 ± 0.02	1.19 ± 0.01	1.3 ± 0.01	0.97 ± 0.01	1.08 ± 0.02
Avg. accuracy	0.17 ± 0.01	0.27 ± 0.01	0.5 ± 0.01	0.47 ± 0.02	0.58 ± 0.01	0.52 ± 0.02

scales as $\mathcal{O}((NJ)^3)$ – the number of independent components is typically very small, but nevertheless the multiplicative cost can be unfeasible in the rare case of very large number of components. This is not a big limitation, since this complexity can be reduced to $\mathcal{O}(J^3)$ by fully factorizing the Gaussian approximate posterior – this is the complexity seen typically with sparse GPs, though this may come at the cost of less accurate posteriors. For our experiments this was not necessary, and non-factorized posterior provided better performance. Computational cost can be reduced also by amortizing the inference, though this again comes at the cost of an amortization gap in the lower bound. Another practical challenge is that t-Processes are non-ergodic. Considering a time signal for simplicity, a single infinitely long time-series does not have a marginal following a Student’s-t distribution, since it is simply a GP re-scaled by a single Gamma random variable. This means that multiple samples are needed for estimation and inference.

8 CONCLUSION

We introduced a new non-linear ICA model, tp-NICA, that uses *t*-process independent components to attain general identifiability. Unlike previous non-linear ICA algorithms, our model is designed for spatial, spatio-temporal, and any other data with high-dimensional latent structures. We also introduced a novel variational learning and inference algorithm that is able to estimate the model despite the non-conjugate structure of the t-Process prior. We also establish gp-NICA as a special case of our model but show that its identifiability is more constrained. Our results show the applicability of our approach on spatial and spatio-temporal data. We hope that our approach will inspire many further identifiable models for this type of data which is common in many important real life applications.

Acknowledgments

H.H. received funding from the HIIT - Helsinki Institute for Information Technology. J.S. is supported by the

University of Cambridge Harding Distinguished Postgraduate Scholars Programme. R.E.T. is supported by Google, Amazon, ARM, Improbable, and EPSRC grant EP/T005386/1. A.H. was funded by a Fellow Position from CIFAR and the Academy of Finland (project #330482). The authors wish to thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources, as well as the CSC – IT Center for Science, Finland, for generous computational resources.

References

- Ashman, M., So, J., Tebbutt, W., Fortuin, V., Pearce, M., and Turner, R. E. (2020). Sparse gaussian process variational autoencoders. *arXiv preprint arXiv:2010.10177*.
- Banville, H., Chehab, O., Hyvärinen, A., Engemann, D.-A., and Gramfort, A. (2021). Uncovering the structure of clinical EEG signals with self-supervised learning. *J. Neural Engineering*, 18(046020).
- Belouchrani, A., Meraim, K. A., Cardoso, J.-F., and Moulines, E. (1997). A blind source separation technique based on second order statistics. *IEEE Trans. on Signal Processing*, 45(2):434–444.
- Casale, F. P., Dalca, A., Saglietti, L., Listgarten, J., and Fusi, N. (2018). Gaussian process prior variational autoencoders. *Advances in neural information processing systems*, 31.
- Crouse, D. F. (2016). On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.
- Fernandez, C. and Steel, M. F. J. (1999). Multivariate student-t regression models: Pitfalls and inference. *Biometrika*, 86(1):153–167.
- Hälvä, H. and Hyvärinen, A. (2020). Hidden Markov nonlinear ICA: Unsupervised learning from nonstationary time series. In *Proc. 36th Conf. on Uncertainty in Artificial Intelligence (UAI2020)*, Toronto, Canada (virtual).

- Hälvä, H., Le Corff, S., Lehéricy, L., So, J., Zhu, Y., Gassiat, E., and Hyvarinen, A. (2021). Disentangling identifiable features from noisy data with structured nonlinear ica. *Advances in Neural Information Processing Systems*, 34:1624–1633.
- Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, page 282. Citeseer.
- Hensman, J., Matthews, A., and Ghahramani, Z. (2015). Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR.
- Hyvärinen, A., Hurri, J., and Hoyer, P. O. (2009). *Natural image statistics: A probabilistic approach to early computational vision.*, volume 39. Springer Science & Business Media.
- Hyvärinen, A., Khemakhem, I., and Morioka, H. (2023). Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning. *arXiv preprint arXiv:2303.16535*.
- Hyvärinen, A. and Morioka, H. (2016). Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems (NIPS2016)*, Barcelona, Spain.
- Hyvärinen, A. and Morioka, H. (2017). Nonlinear ICA of temporally dependent stationary sources. In *Proc. Artificial Intelligence and Statistics (AISTATS2017)*, Fort Lauderdale, Florida.
- Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439.
- Hyvärinen, A., Sasaki, H., and Turner, R. (2019). Nonlinear ICA using auxiliary variables and generalized contrastive learning. In *Proc. Artificial Intelligence and Statistics (AISTATS2019)*, Okinawa, Japan.
- Jazbec, M., Ashman, M., Fortuin, V., Pearce, M., Mandt, S., and Rätsch, G. (2021). Scalable gaussian process variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 3511–3519. PMLR.
- Johnson, M. J., Duvenaud, D. K., Wiltchko, A., Adams, R. P., and Datta, S. R. (2016). Composing graphical models with neural networks for structured representations and fast inference. *Advances in neural information processing systems*, 29.
- Khemakhem, I., Kingma, D. P., Monti, R. P., and Hyvärinen, A. (2020a). Variational autoencoders and nonlinear ICA: A unifying framework. In *Proc. Artificial Intelligence and Statistics (AISTATS2020)*.
- Khemakhem, I., Monti, R. P., Kingma, D. P., and Hyvärinen, A. (2020b). ICE-BeeM: Identifiable conditional energy-based deep models based on nonlinear ICA. In *Advances in Neural Information Processing Systems (NeurIPS2020)*, Virtual.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Klindt, D. A., Schott, L., Sharma, Y., Ustyuzhaninov, I., Brendel, W., Bethge, M., and Paiton, D. M. (2021). Towards nonlinear disentanglement in natural data with temporal sparse coding. In *9th International Conference on Learning Representations, ICLR 2021*.
- Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research.
- Monti, R. P., Zhang, K., and Hyvärinen, A. (2019). Causal discovery with general non-linear relationships using non-linear ICA. In *Proc. 35th Conf. on Uncertainty in Artificial Intelligence (UAI2019)*, Tel Aviv, Israel.
- Pearce, M. (2020). The gaussian process prior vae for interpretable latent dynamics from pixels. In *Symposium on advances in approximate bayesian inference*, pages 1–12. PMLR.
- Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(65):1939–1959.
- RadiantEarthFoundation (2020). CV4A competition Kenya crop type dataset. https://mlhub.earth/data/ref_african_crops_kenya_02.
- Ranganath, R., Gerrish, S., and Blei, D. (2014). Black Box Variational Inference. In Kaski, S. and Corander, J., editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland. PMLR.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.
- Schell, A. and Oberhauser, H. (2023). Nonlinear independent component analysis for discrete-time and continuous-time signals. *Annals of Statistics*. In press.

- Shah, A., Wilson, A., and Ghahramani, Z. (2014). Student-t Processes as Alternatives to Gaussian Processes. In Kaski, S. and Corander, J., editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 877–885, Reykjavik, Iceland. PMLR.
- Snelson, E. and Ghahramani, Z. (2005). Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press.
- Tipping, M. E. and Lawrence, N. D. (2005). Variational inference for student-t models: Robust bayesian interpolation and generalised component analysis. *Neurocomputing*, 69(1-3):123–141.
- Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA. PMLR.
- Wu, P. and Fukumizu, K. (2020). Causal mosaic: Cause-effect inference via nonlinear ica and ensemble method. In *International Conference on Artificial Intelligence and Statistics*, pages 1157–1167. PMLR.
- Yu, S., Tresp, V., and Yu, K. (2007). Robust multi-task learning with t -processes. In *Proceedings of the 24th International Conference on Machine Learning - ICML '07*, pages 1103–1110, Corvallis, Oregon. ACM Press.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
 - (b) Complete proofs of all theoretical results. [Yes]
 - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes]
 - (b) The license information of the assets, if applicable. [Yes]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
 - (d) Information about consent from data providers/curators. [Yes]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials for

Identifiable Feature Learning for Spatial Data with Nonlinear ICA

A Appendix: Algorithm Details

A.1 Pseudo-points

Modeling the latent components as GPs and TPs comes with the challenge of cubic computational cost – that is if each component is distributed over m index locations, then inverting the $m \times m$ GP/TP kernel $\mathbf{K}_{\mathbf{s},\mathbf{s}}$ matrix scales $\mathcal{O}(m^3)$. Pseudo-points can be introduced to reduce this computational complexity (Snelson and Ghahramani, 2005; Quiñero-Candela and Rasmussen, 2005; Hensman et al., 2013, 2015). In essence, the pseudo-points are additional index locations at which we assume further latent variables to exist so that the distribution of all the latents including the pseudo latents \mathbf{u} now becomes:

$$(\mathbf{s}, \mathbf{u}) \sim \mathcal{N} \left(\mathbf{0}, \begin{pmatrix} \mathbf{K}_{\mathbf{s},\mathbf{s}'} & \mathbf{K}_{\mathbf{s},\mathbf{u}'} \\ \mathbf{K}_{\mathbf{u},\mathbf{s}'} & \mathbf{K}_{\mathbf{u},\mathbf{u}'} \end{pmatrix} \right),$$

where the dimension of \mathbf{u} is $J \ll m$ i.e. we assume pseudo-latents at J index locations (the locations are optimized during training). The trick is then that in the likelihood, and its lower-bound, observations will be computed conditional on the \mathbf{u} and then the dependence to the true latents will be managed by the off-diagonal $\mathbf{K}_{\mathbf{s},\mathbf{u}'}$ block. The full covariance $\mathbf{K}_{\mathbf{s},\mathbf{s}}$ is never inverted. This will be detailed in the below derivation.

A.2 ELBO Derivation

Joint pdf with pseudo-points \mathbf{u} :

$$p(\mathbf{x}, \mathbf{s}, \mathbf{u}, \boldsymbol{\tau}) = \prod_{l=1}^m p(\mathbf{x}_l | \mathbf{s}_l; \mathbf{f}) \prod_{i=1}^N p(\mathbf{s}^{(i)}, \mathbf{u}^{(i)} | \tau^{(i)}; \boldsymbol{\theta}^{(i)}) p(\tau^{(i)}; \boldsymbol{\theta}^{(i)}), \quad (9)$$

where for notational simplicity we have written \mathbf{s}_l instead of \mathbf{s}_{l_j} in the first product; also $\mathbf{s}_l = (s_l^{(1)}, \dots, s_l^{(N)})$ and $\mathbf{s}^{(i)} = (s_{l_1}^{(i)}, \dots, s_{l_m}^{(i)})$. The marginal likelihood:

$$\mathcal{L} := \log p(\mathbf{x}) = \log \int_{\boldsymbol{\tau}} \int_{\mathbf{s}} \int_{\mathbf{u}} p(\mathbf{x}, \mathbf{s}, \mathbf{u}, \boldsymbol{\tau}) \, d\mathbf{u} \, d\mathbf{s} \, d\boldsymbol{\tau}, \quad (10)$$

is intractable but can be lower-bounded:

$$\begin{aligned}
 \mathcal{L} &\geq \mathbb{E}_{q(\mathbf{s}, \mathbf{u}, \boldsymbol{\tau})} \left[\log \frac{p(\mathbf{x}, \mathbf{s}, \mathbf{u} \mid \boldsymbol{\tau}) p(\boldsymbol{\tau})}{q(\mathbf{s}, \mathbf{u}, \boldsymbol{\tau})} \right] \\
 &= \mathbb{E}_{q(\boldsymbol{\tau})} \left[\mathbb{E}_{q(\mathbf{s}, \mathbf{u} \mid \boldsymbol{\tau})} \left[\log \frac{p(\mathbf{x}, \mathbf{s}, \mathbf{u} \mid \boldsymbol{\tau})}{q(\mathbf{s}, \mathbf{u} \mid \boldsymbol{\tau})} \right] \right] - \text{KL}[q(\boldsymbol{\tau}) \parallel p(\boldsymbol{\tau})] \\
 &= \mathbb{E}_{q(\boldsymbol{\tau})} \left[\mathbb{E}_{q(\mathbf{s}, \mathbf{u} \mid \boldsymbol{\tau})} [\log p(\mathbf{x} \mid \mathbf{s}, \mathbf{u})] - \text{KL}[q(\mathbf{s}, \mathbf{u} \mid \boldsymbol{\tau}) \parallel p(\mathbf{s}, \mathbf{u} \mid \boldsymbol{\tau})] \right] - \text{KL}[q(\boldsymbol{\tau}) \parallel p(\boldsymbol{\tau})] \\
 &\text{choosing to approximate } q(\mathbf{s} \mid \mathbf{u}, \boldsymbol{\tau}) := p(\mathbf{s} \mid \mathbf{u}, \boldsymbol{\tau}) \text{ results in a further lower bound:} \\
 &\geq \mathbb{E}_{q(\boldsymbol{\tau})} \left[\mathbb{E}_{q(\mathbf{u} \mid \boldsymbol{\tau})} \left[\mathbb{E}_{p(\mathbf{s} \mid \mathbf{u}, \boldsymbol{\tau})} [\log p(\mathbf{x} \mid \mathbf{s})] \right] - \text{KL}[q(\mathbf{u} \mid \boldsymbol{\tau}) \parallel p(\mathbf{u} \mid \boldsymbol{\tau})] \right] - \text{KL}[q(\boldsymbol{\tau}) \parallel p(\boldsymbol{\tau})] \\
 &= \mathbb{E}_{q(\boldsymbol{\tau})} \left[\underbrace{\mathbb{E}_{\tilde{q}(\mathbf{s} \mid \boldsymbol{\tau})} [\log p(\mathbf{x} \mid \mathbf{s})]}_{(*)} - \underbrace{\text{KL}[q(\mathbf{u} \mid \boldsymbol{\tau}) \parallel p(\mathbf{u} \mid \boldsymbol{\tau})]}_{(**)} \right] - \underbrace{\text{KL}[q(\boldsymbol{\tau}) \parallel p(\boldsymbol{\tau})]}_{(***)}, \tag{11}
 \end{aligned}$$

with

$$\tilde{q}(\mathbf{s} \mid \boldsymbol{\tau}) := \int p(\mathbf{s} \mid \mathbf{u}, \boldsymbol{\tau}) q(\mathbf{u} \mid \boldsymbol{\tau}) d\mathbf{u} \tag{12}$$

A.3 Distribution of Latent Components

Note that the pseudo-points are like additional latent variables at some extra locations, and therefore we have:

$$\begin{aligned}
 p(\mathbf{s}, \mathbf{u} \mid \boldsymbol{\tau}) &= \prod_{i=1}^N p(\mathbf{s}^{(i)}, \mathbf{u}^{(i)} \mid \boldsymbol{\tau}^{(i)}; \boldsymbol{\theta}^{(i)}) \\
 &= \prod_{i=1}^N (2\pi)^{-\frac{m+J}{2}} |\mathbf{K}_{\mathbf{r}}^{(i)}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{r}^{(i)})^\top \mathbf{K}_{\mathbf{r}}^{(i)-1} \mathbf{r}^{(i)} \right\}, \tag{13}
 \end{aligned}$$

where

$$\mathbf{r}^{(i)} = \begin{pmatrix} \mathbf{s}^{(i)} \\ \mathbf{u}^{(i)} \end{pmatrix}, \mathbf{K}_{\mathbf{r}}^{(i)} = \begin{pmatrix} \mathbf{K}_{\mathbf{s}, \mathbf{s}'}^{(i)} & \mathbf{K}_{\mathbf{s}, \mathbf{u}'}^{(i)} \\ \mathbf{K}_{\mathbf{u}, \mathbf{s}'}^{(i)} & \mathbf{K}_{\mathbf{u}, \mathbf{u}'}^{(i)} \end{pmatrix},$$

and, for brevity, we define $\mathbf{K}^{(i)} := \frac{\mathbf{K}_{\boldsymbol{\theta}^{(i)}}}{\boldsymbol{\tau}^{(i)}}$. We will re-order elements for later use; that is we define $\mathbf{r} := (\mathbf{s}_1, \dots, \mathbf{s}_m, \mathbf{u}_1, \dots, \mathbf{u}_J) = (\mathbf{s}, \mathbf{u})$, which results in a covariance matrix:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathbf{K}_{\mathbf{s}, \mathbf{s}'} & \mathbf{K}_{\mathbf{s}, \mathbf{u}'} \\ \mathbf{K}_{\mathbf{u}, \mathbf{s}'} & \mathbf{K}_{\mathbf{u}, \mathbf{u}'} \end{pmatrix}, \tag{14}$$

where $\mathbf{K}_{\mathbf{s}, \mathbf{s}'}$ is $mN \times mN$, $\mathbf{K}_{\mathbf{u}, \mathbf{u}'}$ is $JN \times JN$, and $\mathbf{K}_{\mathbf{s}, \mathbf{u}'}$ is $mN \times JN$. These block matrices also have a specific form; using $\mathbf{K}_{\mathbf{s}, \mathbf{s}'}$ as an example, this block can be considered as $m \times m$ matrix of $N \times N$ blocks as its elements where each of the blocks is a diagonal matrix since the latent variables factorize over N . In particular, the form is (dropping subscripts for convenience):

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{1,1} & \cdots & \mathbf{K}_{1,m} \\ \vdots & \ddots & \\ \mathbf{K}_{m,1} & & \mathbf{K}_{m,m} \end{pmatrix}, \tag{15}$$

where $\mathbf{K}_{i,j} = \text{diag}(k_{i,j}^{(1)}, \dots, k_{i,j}^{(N)})$. We can then write the joint as a big block-diagonal Gaussian:

$$p(\mathbf{s}, \mathbf{u} \mid \boldsymbol{\tau}) = (2\pi)^{-\frac{N(m+J)}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \mathbf{r}^\top \boldsymbol{\Sigma}^{-1} \mathbf{r} \right\}. \quad (16)$$

Using the properties of partitioned Gaussians gives the conditional distribution as:

$$p(\mathbf{s} \mid \mathbf{u}, \boldsymbol{\tau}) \propto \exp \left\{ (\mathbf{s} - \boldsymbol{\mu}_{\mathbf{s}|\mathbf{u}})^\top \boldsymbol{\Sigma}_{\mathbf{s}|\mathbf{u}}^{-1} (\mathbf{s} - \boldsymbol{\mu}_{\mathbf{s}|\mathbf{u}}) \right\}, \quad (17)$$

where

$$\boldsymbol{\mu}_{\mathbf{s}|\mathbf{u}} = \mathbf{K}_{\mathbf{s},\mathbf{u}'} \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} \mathbf{u} \quad (18)$$

$$\boldsymbol{\Sigma}_{\mathbf{s}|\mathbf{u}} = \mathbf{K}_{\mathbf{s},\mathbf{s}'} - \mathbf{K}_{\mathbf{s},\mathbf{u}'} \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} \mathbf{K}_{\mathbf{s},\mathbf{u}'}^\top, \quad (19)$$

And the marginal distribution:

$$p(\mathbf{u} \mid \boldsymbol{\tau}) \propto \exp \left\{ \mathbf{u}^\top \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} \mathbf{u} \right\}. \quad (20)$$

A.4 Approximate Posterior

We assume following form of the approximate posterior:

$$\begin{aligned} q(\mathbf{u}, \boldsymbol{\tau}) &= q(\mathbf{u} \mid \boldsymbol{\tau}; \boldsymbol{\phi}, \boldsymbol{\theta}) q(\boldsymbol{\tau}; \boldsymbol{\lambda}) \\ &:= \frac{\prod_{j=1}^J \psi(\mathbf{u}_j; \boldsymbol{\phi}_j) \prod_{i=1}^N p(\mathbf{u}^{(i)} \mid \tau^{(i)}; \boldsymbol{\theta}^{(i)})}{Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\tau})} \prod_{i=1}^N q(\tau^{(i)}; \lambda^{(i)}), \end{aligned} \quad (21)$$

where J is the number of pseudo-locations, $\mathbf{u}_j = (u_j^{(i)})_{i=1:N}$, and $\mathbf{u}^{(i)} = (u_j^{(i)})_{j=1:J}$. The approximate likelihood is assumed to be a Gaussian factor:

$$\prod_{j=1}^J \psi(\mathbf{u}_j; \boldsymbol{\phi}_j) = \prod_{j=1}^J \exp \left\{ -\frac{1}{2} (\mathbf{u}_j^\top \tilde{\mathbf{W}}_j^\top \tilde{\mathbf{W}}_j \mathbf{u}_j - 2\mathbf{u}_j^\top \tilde{\mathbf{W}}_j^\top \tilde{\mathbf{y}}_j) \right\}. \quad (22)$$

Now define:

$$\begin{aligned} \boldsymbol{\Lambda}_j &= \tilde{\mathbf{W}}_j^\top \tilde{\mathbf{W}}_j, \\ \mathbf{m}_j &= \tilde{\mathbf{W}}_j^\top \tilde{\mathbf{y}}_j, \end{aligned} \quad (23)$$

Next, recall $\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_J)$, and define a block-diagonal matrix $\boldsymbol{\Lambda} := \text{diag}(\boldsymbol{\Lambda}_1, \dots, \boldsymbol{\Lambda}_J)$, which allows us to simplify above into:

$$\prod_{j=1}^J \psi(\mathbf{u}_j; \boldsymbol{\phi}_j) = \exp \left\{ -\frac{1}{2} (\mathbf{u}^\top \boldsymbol{\Lambda} \mathbf{u} - 2\mathbf{u}^\top \mathbf{m}) \right\}. \quad (24)$$

With (24) and (20), we thus have:

$$\begin{aligned} q(\mathbf{u} \mid \boldsymbol{\tau}; \boldsymbol{\phi}, \boldsymbol{\theta}) &\propto \exp \left\{ -\frac{1}{2} (\mathbf{u}^\top \boldsymbol{\Lambda} \mathbf{u} - 2\mathbf{u}^\top \mathbf{m}) \right\} (2\pi)^{-\frac{NJ}{2}} |\mathbf{K}_{\mathbf{u},\mathbf{u}'}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \mathbf{u}^\top \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} \mathbf{u} \right\} \\ &= (2\pi)^{-\frac{NJ}{2}} |\mathbf{K}_{\mathbf{u},\mathbf{u}'}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{u}^\top (\boldsymbol{\Lambda} + \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1}) \mathbf{u} - 2\mathbf{u}^\top \mathbf{m}) \right\}. \end{aligned}$$

Simplify with $\mathbf{J} = (\mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} + \mathbf{\Lambda})$ and $\mathbf{h} = \mathbf{J}^{-1}\mathbf{m}$:

$$q(\mathbf{u} \mid \boldsymbol{\tau}; \boldsymbol{\phi}, \boldsymbol{\theta}) \propto (2\pi)^{-\frac{NJ}{2}} |\mathbf{K}_{\mathbf{u},\mathbf{u}'}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{u}^\top \mathbf{J} \mathbf{u} - 2\mathbf{u}^\top \mathbf{J} \mathbf{h}) \right\}. \quad (25)$$

The normalizer can be calculated as:

$$\begin{aligned} Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\tau}) &= (2\pi)^{-\frac{NJ}{2}} |\mathbf{K}_{\mathbf{u},\mathbf{u}'}|^{-\frac{1}{2}} \int \exp \left\{ -\frac{1}{2} (\mathbf{u}^\top \mathbf{J} \mathbf{u} - 2\mathbf{u}^\top \mathbf{J} \mathbf{h}) \right\} d\mathbf{u} \\ &= (2\pi)^{-\frac{NJ}{2}} |\mathbf{K}_{\mathbf{u},\mathbf{u}'}|^{-\frac{1}{2}} (2\pi)^{\frac{NJ}{2}} |\mathbf{J}^{-1}|^{\frac{1}{2}} \exp \left\{ \frac{1}{2} \mathbf{h}^\top \mathbf{J} \mathbf{h} \right\} \\ &= |\mathbf{K}_{\mathbf{u},\mathbf{u}'}|^{-\frac{1}{2}} |\mathbf{J}|^{-\frac{1}{2}} \exp \left\{ \frac{1}{2} \mathbf{m}^\top \mathbf{J}^{-1} \mathbf{m} \right\} \\ &= |\mathbf{\Lambda} \mathbf{K}_{\mathbf{u},\mathbf{u}'} + \mathbf{I}|^{-\frac{1}{2}} \exp \left\{ \frac{1}{2} \mathbf{m}^\top \mathbf{J}^{-1} \mathbf{m} \right\} \\ \Rightarrow \log Z &= \frac{1}{2} \mathbf{m}^\top \mathbf{K}_{\mathbf{u},\mathbf{u}'} (\mathbf{\Lambda} \mathbf{K}_{\mathbf{u},\mathbf{u}'} + \mathbf{I})^{-1} \mathbf{m} - \frac{1}{2} \log |\mathbf{\Lambda} \mathbf{K}_{\mathbf{u},\mathbf{u}'} + \mathbf{I}| \end{aligned} \quad (26)$$

And for completeness:

$$q(\mathbf{u} \mid \boldsymbol{\tau}; \boldsymbol{\phi}, \boldsymbol{\theta}) = |2\pi \mathbf{J}^{-1}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{u} - \mathbf{h})^\top \mathbf{J} (\mathbf{u} - \mathbf{h}) \right\}. \quad (27)$$

A.5 Computing (*)

We have:

$$\tilde{q}(\mathbf{s} \mid \boldsymbol{\tau}) := \int p(\mathbf{s} \mid \mathbf{u}, \boldsymbol{\tau}) q(\mathbf{u} \mid \boldsymbol{\tau}) d\mathbf{u}. \quad (28)$$

Substitute in (17) and (27):

$$\tilde{q}(\mathbf{s} \mid \boldsymbol{\tau}) \propto \int \exp \left\{ (\mathbf{s} - \boldsymbol{\mu}_{\mathbf{s}|\mathbf{u}})^\top \boldsymbol{\Sigma}_{\mathbf{s}|\mathbf{u}}^{-1} (\mathbf{s} - \boldsymbol{\mu}_{\mathbf{s}|\mathbf{u}}) \right\} \exp \left\{ -\frac{1}{2} (\mathbf{u} - \mathbf{h})^\top \mathbf{J} (\mathbf{u} - \mathbf{h}) \right\} d\mathbf{u}, \quad (29)$$

so that the *approximate* marginal is a Gaussian:

$$\tilde{q}(\mathbf{s} \mid \boldsymbol{\tau}) \propto \exp \left\{ (\mathbf{s} - \tilde{\boldsymbol{\mu}})^\top \tilde{\boldsymbol{\Sigma}}^{-1} (\mathbf{s} - \tilde{\boldsymbol{\mu}}) \right\}, \quad (30)$$

with:

$$\begin{aligned} \tilde{\boldsymbol{\mu}} &= \mathbf{K}_{\mathbf{s},\mathbf{u}'} \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} \mathbf{h} \\ &= \mathbf{K}_{\mathbf{s},\mathbf{u}'} \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} (\mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} + \mathbf{\Lambda})^{-1} \mathbf{m} \\ &= \mathbf{K}_{\mathbf{s},\mathbf{u}'} (\mathbf{K}_{\mathbf{u},\mathbf{u}'} + \mathbf{\Lambda}^{-1})^{-1} \mathbf{\Lambda}^{-1} \mathbf{m} \\ &= \mathbf{K}_{\mathbf{s},\mathbf{u}'} (\mathbf{I} + \mathbf{\Lambda} \mathbf{K}_{\mathbf{u},\mathbf{u}'})^{-1} \mathbf{m} \end{aligned} \quad (31)$$

$$\begin{aligned} \tilde{\boldsymbol{\Sigma}} &= \mathbf{K}_{\mathbf{s},\mathbf{s}'} - \mathbf{K}_{\mathbf{s},\mathbf{u}'} \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} \mathbf{K}_{\mathbf{u},\mathbf{s}} + \mathbf{K}_{\mathbf{s},\mathbf{u}'} \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} \mathbf{J}^{-1} \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} \mathbf{K}_{\mathbf{u},\mathbf{s}'} \\ &= \mathbf{K}_{\mathbf{s},\mathbf{s}'} - \mathbf{K}_{\mathbf{s},\mathbf{u}'} (\mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} - \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} \mathbf{J}^{-1} \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1}) \mathbf{K}_{\mathbf{u},\mathbf{s}'} \\ &= \mathbf{K}_{\mathbf{s},\mathbf{s}'} - \mathbf{K}_{\mathbf{s},\mathbf{u}'} (\mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} - \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1} (\mathbf{\Lambda} + \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1})^{-1} \mathbf{K}_{\mathbf{u},\mathbf{u}'}^{-1}) \mathbf{K}_{\mathbf{u},\mathbf{s}'} \\ &= \mathbf{K}_{\mathbf{s},\mathbf{s}'} - \mathbf{K}_{\mathbf{s},\mathbf{u}'} (\mathbf{K}_{\mathbf{u},\mathbf{u}'} + \mathbf{\Lambda}^{-1})^{-1} \mathbf{K}_{\mathbf{u},\mathbf{s}'} \\ &= \mathbf{K}_{\mathbf{s},\mathbf{s}'} - \mathbf{K}_{\mathbf{s},\mathbf{u}'} (\mathbf{I} + \mathbf{\Lambda} \mathbf{K}_{\mathbf{u},\mathbf{u}'})^{-1} \mathbf{\Lambda} \mathbf{K}_{\mathbf{u},\mathbf{s}'} \end{aligned} \quad (32)$$

We can sample from this distribution to approximate (*) and take gradients using the reparametrization trick. Further, note that in practice we only need the m , $N \times N$, diagonal blocks of the above covariance matrix as we only have to sample from $\tilde{q}(\mathbf{s}_t | \boldsymbol{\tau}) \forall t$, rather than the full joint.

A.6 Computing (**)

$$\text{KL}[q(\mathbf{u} | \boldsymbol{\tau}) \| p(\mathbf{u} | \boldsymbol{\tau})] = \int_{\mathbf{u}} q(\mathbf{u} | \boldsymbol{\tau}) \log \frac{q(\mathbf{u} | \boldsymbol{\tau})}{p(\mathbf{u} | \boldsymbol{\tau})} \quad (33)$$

$$= \int_{\mathbf{u}} q(\mathbf{u} | \boldsymbol{\tau}) \log \frac{\prod_{j=1}^J \psi(\mathbf{u}_j; \boldsymbol{\phi}_j) \prod_{i=1}^N p(\mathbf{u}^{(i)} | \tau^{(i)}; \boldsymbol{\theta}^{(i)})}{Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\tau}) \prod_{i=1}^N p(\mathbf{u}^{(i)} | \tau^{(i)}; \boldsymbol{\theta}^{(i)})} \quad (34)$$

$$= \sum_{j=1}^J \mathbb{E}_{q(\mathbf{u}_j | \boldsymbol{\tau})} [\log \psi(\mathbf{u}_j; \boldsymbol{\phi}_j)] - \log Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\tau}), \quad (35)$$

where $\log Z(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\theta}, \boldsymbol{\tau})$ can be computed using (26), while:

$$\begin{aligned} \sum_{j=1}^J \mathbb{E}_{q(\mathbf{u}_j | \boldsymbol{\tau})} [\log \psi(\mathbf{u}_j; \boldsymbol{\phi}_j)] &= \sum_{j=1}^J \mathbb{E}_{q(\mathbf{u}_j | \boldsymbol{\tau})} \left[-\frac{1}{2} \mathbf{u}_j^\top \boldsymbol{\Lambda}_j \mathbf{u}_j + \mathbf{u}_j^\top \mathbf{m}_j \right] \\ &= \sum_{j=1}^J -\frac{1}{2} \text{Tr}(\boldsymbol{\Lambda}_j [\mathbf{J}^{-1}]_j) - \frac{1}{2} \mathbf{h}_j^\top \boldsymbol{\Lambda}_j \mathbf{h}_j + \mathbf{h}_j^\top \mathbf{m}_j \\ &= \sum_{j=1}^J -\frac{1}{2} \text{Tr}(\boldsymbol{\Lambda}_j [\mathbf{K}_{\mathbf{u}, \mathbf{u}} (\boldsymbol{\Lambda} \mathbf{K}_{\mathbf{u}, \mathbf{u}} + \mathbf{I})^{-1}]_j) - \frac{1}{2} \mathbf{h}_j^\top \boldsymbol{\Lambda}_j \mathbf{h}_j + \mathbf{h}_j^\top \mathbf{m}_j \\ &= \sum_{j=1}^J -\frac{1}{2} \text{Tr}([\mathbf{K}_{\mathbf{u}, \mathbf{u}} (\boldsymbol{\Lambda} \mathbf{K}_{\mathbf{u}, \mathbf{u}} + \mathbf{I})^{-1} \boldsymbol{\Lambda}]_j) - \frac{1}{2} \mathbf{h}_j^\top \boldsymbol{\Lambda}_j \mathbf{h}_j + \mathbf{h}_j^\top \mathbf{m}_j \end{aligned}$$

A.7 Computing (***)

We will make $\text{KL}[q(\boldsymbol{\tau}) \| p(\boldsymbol{\tau})]$ tractable by assuming $q(\boldsymbol{\tau})$ to follow factorial gamma distribution and using properties of exponential families:

$$\text{KL}[q(\boldsymbol{\tau}) \| p(\boldsymbol{\tau})] = \sum_{i=1}^N \text{KL}[q(\tau^{(i)}; \lambda^{(i)}) \| p(\tau^{(i)}; \gamma^{(i)})] \quad (36)$$

$$= \sum_{i=1}^N A(\boldsymbol{\eta}_p^{(i)}) - A(\boldsymbol{\eta}_q^{(i)}) - (\boldsymbol{\eta}_p^{(i)} - \boldsymbol{\eta}_q^{(i)})^\top \nabla A(\boldsymbol{\eta}_q^{(i)}), \quad (37)$$

where $\boldsymbol{\eta}^{(i)}$ are the natural parameters, and $A(\cdot)$ is the log-normalizer and gradient of the log normalizer gives the mean parameters.

B Appendix: Proofs of Theorems

Proof of Theorem 1 (A1) requires the noise-free data $\mathbf{f}(\mathbf{s})$ to have tails that are only slightly heavier than Gaussian, while (A2) and (A3) ensure non-degeneracy and non-Gaussianity of the noise-free data, respectively. With these assumptions, Theorem 1 of Hälvä et al. (2021) shows that one can identify the noise-free mixture from the noisy mixture: $p(\mathbf{x}) = \tilde{p}(\mathbf{x}) \implies p(\mathbf{f}(\mathbf{s})) = p(\tilde{\mathbf{f}}(\mathbf{s}))$. For our tp-NICA model, the noise-free distribution can be written as (assuming zero-mean TP):

$$\prod_{j=1}^m |\mathbf{J}\mathbf{g}(\mathbf{x}_l)| \prod_{i=1}^N \pi^{-\frac{d}{2}} |\boldsymbol{\Sigma}^{(i)}|^{-\frac{1}{2}} \nu^{(i)\frac{\nu^{(i)}}{2}} \frac{\Gamma(\frac{\nu^{(i)}+d}{2})}{\Gamma(\frac{\nu^{(i)}}{2})} (\nu^{(i)} + \mathbf{g}^{(i)}(\mathbf{x}_{1:m})^T \boldsymbol{\Sigma}^{-1,(i)} \mathbf{g}^{(i)}(\mathbf{x}_{1:m}))^{-\frac{\nu^{(i)}+d}{2}}, \quad (38)$$

where \mathbf{J} is the Jacobian resulting from the change of variable, and $\mathbf{g} = (\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(N)}) = \mathbf{f}^{-1}$. In other words, $\mathbf{g}^{(i)}(\mathbf{x}_{1:m}) = \mathbf{s}^{(i)} = (s_1^{(i)}, \dots, s_m^{(i)})$, and the log-likelihood becomes:

$$\log p(\mathbf{x}) = \sum_{j=1}^m \log |\mathbf{J}\mathbf{g}(\mathbf{x}_l)| - \sum_{i=1}^N \frac{\nu^{(i)} + d}{2} \log (\nu^{(i)} + \mathbf{s}^{(i)T} \boldsymbol{\Sigma}^{-1,(i)} \mathbf{s}^{(i)}) + \text{const}. \quad (39)$$

We drop the superscript (i) for brevity and define $Q := \log p$. Also we will use $\mathbf{A}_{:,i}$ to denote the i -th column of a matrix, and similar notation for rows.

$$\frac{\partial}{\partial s_{l_k}} Q = -\frac{\nu + d}{(\nu + \mathbf{s}^T \boldsymbol{\Sigma}^{-1} \mathbf{s})} \mathbf{s}^T \boldsymbol{\Sigma}_{:,k}^{-1} \quad (40)$$

$$\begin{aligned} \frac{\partial^2}{\partial s_{l_k} \partial s_{l_j}} Q &= -\frac{\nu + d}{(\nu + \mathbf{s}^T \boldsymbol{\Sigma}^{-1} \mathbf{s})} \boldsymbol{\Sigma}_{j,k}^{-1} \\ &+ 2 \frac{\nu + d}{(\nu + \mathbf{s}^T \boldsymbol{\Sigma}^{-1} \mathbf{s})^2} \mathbf{s}^T \boldsymbol{\Sigma}_{:,k}^{-1} \mathbf{s}^T \boldsymbol{\Sigma}_{:,l}^{-1}. \end{aligned} \quad (41)$$

The s_{l_k}, s_{l_j} don't factorize in above, and thus we can not rewrite this as:

$$\frac{\partial^2}{\partial s_{l_k} \partial s_{l_j}} Q = c \alpha(s_{l_k}, \mathbf{s}_{(-l_k, -l_j)}) \alpha(s_{l_j}, \mathbf{s}_{(-l_k, -l_j)}) \quad (42)$$

In other words, the distribution is not *quasi*-Gaussian. This is precisely the condition needed to satisfy assumptions of Theorem 2 in Hälvä et al. (2021) and by this theorem $p(\mathbf{f}(\mathbf{s})) = p(\tilde{\mathbf{f}}(\mathbf{s})) \implies \mathbf{f}^{-1} \sim \tilde{\mathbf{f}}^{-1}$, where \sim denotes equivalence up to permutation and coordinate-wise bijective transformation of the elements of the de-mixing function \mathbf{f}^{-1} .

Proof of Theorem 2 Theorem 5 in Hälvä et al. (2021) shows that if the covariance kernel functions of the components are distinct, the components are identifiable (hence sufficiency is already satisfied). It remains to prove necessity. To do this, it is enough to show that the condition is necessary for any specific \mathbf{f} – indeed it is enough to show the necessity for linear ICA, which we do by adapting its well-known theory (Belouchrani et al., 1997). We proceed by contradiction: Assume there exist an identifiable model with a linear mixing function and components with non-distinct kernels $\kappa^{(i)} = \kappa^{(j)}$, that is, some two components i and j have the same covariance kernel. Next, consider the alternative components obtained by an orthogonal rotation: $s'^{(i)} = \frac{1}{\sqrt{2}}[s^{(i)} + s^{(j)}]$ and $s'^{(j)} = \frac{1}{\sqrt{2}}[s^{(i)} - s^{(j)}]$. These new components have the same autocorrelations as the original components. They are also uncorrelated, and hence independent. The Jacobian of the orthogonal transformation is identity so, the likelihood of these

components is equal to the likelihood of the original components. This implies unidentifiability of the components. Distinct kernels are therefore necessary in this specific example. Thus, if distinctness of kernels functions is violated, nonlinear mixing cannot be identifiable in general.

C Appendix: Experiment Details

Experiments on Simulated Data In total 1024 independent samples were used to train each model specification, that is the training data had 1024 samples of $M \times 32 \times 32$ observations where each was generated by a nonlinear mixing matrix of $N \times 32 \times 32$ independent components, where M and N are the data dimension and the number of independent components. Output noise was added to the observations such that it accounted to approximately 10% of the latent components overall variance. The training of these samples was done in minibatches of 8 samples. This entire procedure was repeated 6 times from different random seeds to ensure reproducibility, the variance between the seeds' results is depicted by error bars in the Figures 2. In these simulations the d.o.f. of the tp-NICA was set to four to ensure fat tails and thus ensure that the data generated is different from gp-NICA. Estimation of the d.o.f. parameter, ν , is in general known to be potentially ill-defined (Fernandez and Steel, 1999) and thus we set ν to the ground-truth values in the simulation as it was not of major focus in our paper (though in our experience, estimating it resulted nearly identical results). All other parameters were fully estimated. Adam optimizer was used with learning rates of $1e^{-1}$ and $1e^{-4}$ for the variational parameters and model parameters respectively. In order to make our code as stable and computationally efficient as possible, matrix inverses were never performed directly, but rather always using linear solves of matrix-vector products. The most important factor in computational efficacy of the model is the setting on number of pseudo-points used. We found that 50-points provided appropriate balance between speed and good identifiability results. Typical training time of our model was around 12-24h on either a single A100 or V100 NVIDIA GPU. All further details can be found in our code at <https://github.com/cambridge-mlg/tp-nica>.

Experiments on CV4A Satellite Data The Computer Vision for Agriculture (CV4A) Kenya Crop Type dataset (RadiantEarthFoundation, 2020) was originally designed for an supervised classification task of different field crops and was published at the ICLR 2020. We adapted this data to our use by considering the data in unsupervised sense after first cropping the satellite images into 32x32 images (here were benefited from the data processing code available at a public github repository that created the crops and normalized the images by their mean values and standard deviation. The 16 dimensional satellite image data from the Sentinel-12 satellite consists of the following bands: ultra blue, blue, red, green, five visible and near infrared wavelengths, four short-wave infrared wavelengths, cloud probability layer, and vegetation index layers constructed from the other layers.

The <https://github.com/cambridge-mlg/tp-nica> repository for our paper also contains all the further details for our experiments on this data.

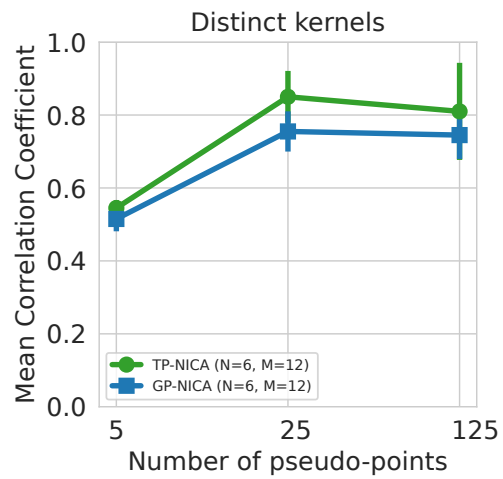


Figure 4: Simulation results for different number of pseudo-points for tp-NICA and gp-NICA models with three mixing layers and assuming distinct kernels.